

Praktikum „Bioinformatics Lab“ SS 2010

Week 2: “Building tarballs using automake and autoconf”

6th of May by Florian Seifert

1. Preparations

After introducing to vim we wrote a short script in perl which removes all space from a textfile and overwrites it. You might have seen advanced vim editors, which had syntax highlighting and multi file support which is not only to programmers useful. In my version I can turn syntax highlighting on with `':syn on'` but I am not sure if this works on every version of vim since I added a few add-ons like vim-nox.

I will just recommend you the Wikipedia page¹ at this point; they provide links including the developer page, a quick reference and even a video tutorial. Another thing beginners might want to try is vim tutor. Simply enter `'vimtutor'` into some command line and an interactive text tutor covering the basics of vim in about 20 – 30 minutes will show up.

In order to get the necessary packages we figured out about *apt-get install NAME* and *aptitude*. The following instructions are derived from the course wiki, *info automake* and some web research²³.

¹ <http://de.wikipedia.org/wiki/Vim>

² <http://www.hs-augsburg.de/~hhoegl/oss/hausarbeiten-04/oss-autotools.pdf>

³ http://www.makelinux.net/books/autobook-1.5/autobook.html#SEC_Top

2. Configuring the build process

Let's start writing a file named 'Configure.ac' which provides basic information about the package to generate. It is used by *autoconf* and *aclocal* to configure the macros that create './configure'-file and should at least contain⁴:

```
AC_INIT ([name of tool], [version], [author])
AM_INIT_AUTOMAKE
AC_CONFIG_FILES ([Makefile])
AC_OUTPUT
```

Next we need the 'Makefile.am' which is used by *automake* to generate the 'Makefile.in' which is the final instruction for the installation.

Usually this contains the name of the program and the sourcecode files. Since we have script which doesn't need to be compiled is sufficient to give some advice on the tarball we want it to belong to.

```
dist_bin_scripts = spacedel
```

Finally we need some additional text files named AUTHORS, ChangeLog, NEWS and README, that can be empty (\$ touch NAME) but are required. On the other hand a readme might not be a bad idea.

3. Letting the tools do their job

Remember by default you will need to be root to install anything. We start with 'aclocal' if we don't have a aclocal.m4 file from a previous try. Then we use 'autoconf' and get the configure file. Before we are able to run this we need to build Makefile.in by using 'automake'. Finally we can type './configure', specify a destination

⁴ http://www.dis.com/gnu/autoconf/configure_002eac-Layout.html#configure_002eac-Layout

folder using `--prefix=/some/folder` and start the install process with `'make'`⁵ and `'make install'`. `'make'` is technically a shell script⁶ that can be used to run any kind of tools and additional parameters very efficiently: it's only compiling what's necessary.

4. Packing the tar ball

Actually it isn't hard anymore if all your files are up to date you simply enter `'make dist'` and receive a `.tar.gz` archive named as defined earlier. `'make distcheck'`⁷ is a useful automatic check whether your archive will technically be working but it's not complaining about missing man page, which should be present.

5. Adding a man page

Linux users are used to have man pages coming along with new commands, so you might want to have one, too. This kind of help can be coded in Perlpod. After installing `'perlpod'` and `'perlpod-doc'` one can convert the output with `'pod2man'` into man page files.

To add these files to the tarball we need to modify the `Makefile.am`

```
dist_bin_scripts = spacedel
#dist_noinst_DATA = spacedel.pod
#man_MANS = spacedel.1
spacedel.1: spacedel.pod
pod2man -c 'User Commands' -r "$(VERSION)" $< $@
clean-local: rm -f spacedel.1
```

After updating the `Makefile` you should run `'autoconf'` again and follow the steps as above to create a tarball.

⁵ <http://linuxwiki.de/make> (including links for further reading)

⁶ <http://www.jfranken.de/homepages/johannes/vortraege/make.de.html>

⁷ http://sourceware.org/autobook/autobook/autobook_102.html

6. Results

Finally we can run our script by simply entering its name and target. For installing a simple script the whole process might be a little uncomfortable but for developers who distribute larger packages that do not have their own system (like java's .jar) it's probably quite useful. Furthermore we learned to use vim a little and might remember make the next time we have a problem with a lot of inter-dependencies.